REMARKS/ARGUMENTS

1. Claims 1-12, 31, and 32 are Patentable Over the Cited Art

The Examiner rejected claims 1-12, 31, and 32 as obvious (35 U.S.C. §103(a)) over Choy (U.S. Patent No. 5,551,027), Cornwell (U.S. Pub. No. 2002/0032678), and SGI (Linux Man page for fetch). Applicants traverse with respect to the amended claims.

Claim 1 recites a method for accessing data in a database table, comprising: receiving a fetch request to fetch data from a base table that satisfies a query predicate, wherein rows of the base table are stored in table partitions and wherein there is one index partition for each determined table partition, wherein each index partition includes nodes, wherein each node in each index partition includes at least one key column value from a corresponding table row in the table partition associated with the index partition and a location identifier identifying the corresponding table row in the corresponding table partition; comparing a direction indicated in the fetch request and an ordering of the index partitions; setting a fetch direction based on a result of the comparison of the direction indicated in the fetch request and the ordering of the index partitions; scanning the index partitions in the fetch direction to determine a set of nodes from the index partitions whose key column value satisfies the query predicate; ordering the set of determined nodes from the index partitions; selecting one node from the ordered set based on a position of the node in the ordering; and returning data from the table row identified by the location identifier in the selected node in response to the fetch request.

The Examiner cited paras. 87, 9, 10, 13, and 44 of Cornwell as teaching the claim requirement of comparing a direction indicated in the fetch request and an ordering of the index partitions. (FOA, pg. 3) Applicants traverse.

The cited para. 87 mentions a FETCH command and returning data at the row at the cursor position in the result table, and whether the returned data should reflect changes made to the base table, such as with a FETCH sensitive. The cited paras. 9, 10, and 13 mention allowing an application to fetch forward or backward from the current position of the result set. A forward only cursor allows the application to fetch forward and a dynamic cursor reflects changes made to the rows in the result table when scrolling through the cursor. The cited para. 44 mentions that a static cursor specified as sensitive indicates that changes made to the underlying base table are provided to the cursor when fetching rows from the result set.

Although the cited Cornwell discusses the operations of a fetch request to fetch data from a result set with a scrollable cursor, there is no teaching or mention of the claim requirement of comparing a direction indicated in the fetch request and an ordering of the index partitions. Cornwell mentions that one may fetch forward or backward from the cursor position in the result set, but the Examiner has not cited where Cornwell teaches that the direction indicated in the fetch request, e.g., forward or backward, id compared to an ordering of index partitions.

The Examiner cited paras. 57, 77, 99, and 100 of Cornwell as teaching the claim requirement of setting a fetch direction based on a result of the comparison of the direction indicated in the fetch request and the ordering of the index partitions and scanning the index partitions in the fetch direction. (FOA, pgs. 4-5) Applicants traverse.

The cited para. 57 mentions that for each column in a select list, a control block is generated for the item in the select list, and a control block is generated for the expression in the select list. Para. 57 further discusses how control blocks are generated for the base table column. The cited para. 77 mentions that after the result table is populated with rows, the application may issue FETCH statements to fetch rows, and how the data may be fetched from the result table (insensitive) or fetched from the base table corresponding to the row in the result table (sensitive). The cited paras. 99-100 discusses a FETCH ABSOLUTE to fetch forward or negative from the first entry in the result table, and how to fetch absolute.

Although the cited Cornwell discusses various fetch operations that may be performed and how to use control blocks, the Examiner has not cited where Cornwell specifically teaches the claim requirement that the fetch direction is set based on the result of comparing the direction indicated in the fetch request and the ordering of the index partitions. In fact, the cited para. 99 mentions that the direction is based on whether the value "k" supplied with the FETCH ABSOLUTE is positive or negative. The Examiner has not cited how using a parameter k with the FETCH command teaches that the fetch direction is based on comparing the direction indicated in the fetch request and the ordering of the index partitions.

Thus, the cited Cornwell does not teach the limitations for which it was cited, and the Examiner has not cited other parts of the references that teach these limitations which the cited Cornwell does not teach for the above discussed reasons.

Accordingly, claim 1 distinguishes over the cited art because the cited combination of references does not teach or suggest all the claim requirements.

Claims 3-12, 31, and 32 are patentable over the cited art because they depend from claim 1, which is patentable over the cited art for the reasons discussed above. Moreover, the below discussed dependent claims provide additional ground of patentability over the cited art.

Claim 3 depends from claim 1 and further recites that the fetch direction is set opposite the direction indicated in the fetch request if the direction indicated in the fetch request is opposite the ordering of the index partitions.

The Examiner cited para. [0099] of Cornwell as teaching the additional requirements of claim 3. (FOA, pg. 4). Applicants traverse.

The cited para. [0099] mentions performing a FETCH ABSOLUTE of k, where k is the number of rows to fetch forward for a +k or negative (-k). The data manager determines the absolute row number of the entry in the result table pointed to by the cursor and the page including the entry, and determines the relative distance of the requested entry from the current entry as the absolute value of K.

Although the cited para. [0099] discusses how to determine the relative distance of a current result table entry to the entry to which the fetch command wants to fetch, the Examiner has not explained where the cited paragraph teaches or suggests setting the fetch direction in which the index partitions are scanned opposite the direction indicated in the fetch request if the ordering of the index partitions is opposite the direction in the fetch request. Instead, the cited para. [0099] discusses determining a relative distance to fetch. This discussion of para. 99 does not teach or suggest determining whether to modify the fetch direction in which the index partitions are scanned based on whether a current fetch direction is opposite an ordering of the index partitions.

Accordingly, claim 3 provides additional grounds of patentability over the cited art because the additional requirements of claim 3 are not taught or suggested in the cited art.

Claim 4 recites that setting the fetch direction comprises: setting the fetch direction to backward if the fetch direction is backward and the fetch direction is not opposite the ordering of the index partitions or if the fetch direction is forward and the fetch direction is opposite the ordering of the index partitions; and setting the fetch direction to forward if the fetch direction is backward and the fetch direction is opposite the ordering of the index partitions or if the fetch direction is forward and the fetch direction is not opposite the ordering of the index partitions.

The Examiner cited para. [0099] of Cornwell as teaching the additional requirements of claim 4. (FOA, pg. 5) Applicants traverse for the following reasons.

The cited para. [0099] discusses a fetch absolute operation, which is the number of rows to fetch forward or backward from the first entry in the result table. This is performed by determining the distance from the current entry to the requested entry, and then convert this command to a fetch relative to fetch to the requested position.

The Examiner has not shown where the cited para. [0099] teaches or suggests setting the fetch direction in which the index is scanned based on the partition index ordering and the direction indicated in the fetch request. Instead, the cited para. [0099] discusses how to fetch to an absolute requested row from the current position, not to change the direction of a fetch based on the index ordering.

Further, the Examiner has not cited any part of Cromwell and paragraphs of setting the fetch direction to backward if the fetch direction is backward and the fetch direction is not opposite the ordering of the index partitions or if the fetch direction is forward and the fetch direction is opposite the ordering of the index partitions and setting the fetch direction to forward if the fetch direction is backward and the fetch direction is opposite the ordering of the index partitions or if the fetch direction is forward and the fetch direction is not opposite the ordering of the index partitions. These specific operations to set the direction based on the fetch direction and the ordering of the index partitions is nowhere taught or suggested in the cited art.

Accordingly, claim 4 provides additional grounds of patentability over the cited art because the additional requirements of these claims are not taught or suggested in the cited art.

Claim 5 depends from claim 1 and further requires that if the fetch request is a first fetch of the fetch request, then selecting one node starting from one of: a lowest key value from each index partition if the fetch direction is forward or highest key value from each index partition if the fetch direction is backward.

The Examiner cited para. [0087] as teaching the additional requirements of these claims. (FOA, pg. 5)

The cited para. [0087] discusses positioning the cursor to the position specified in the fetch, e.g., prior, first, last, current, etc if the fetch is insensitive and then returning the row positioned at the cursor. If the returned row was previously fetched, with a fetch sensitive it would reflect any changes made to the base table prior to the fetch sensitive operation.

Although the cited para. [0087] discusses how to position the cursor on a row depending on the fetch specified, there is no teaching or suggestion of selecting a node from a lowest or highest key value from each index partition depending on the fetch direction, the Examiner has not shown where cited para. [0087] teaches selecting key values from index partitions as claimed. Instead, the cited para. [0087] discusses how to fetch to a requested position and then return the data from the row to which the cursor points, which may comprise changed data if the previous fetch was fetch sensitive.

Accordingly, claim 5 provides additional grounds of patentability over the cited art because the additional requirements of claim 5 are not taught or suggested in the cited art.

Claim 6 depends from claim 1 and further recites if the fetch request is not a first fetch of the fetch request, then determining whether the fetch direction in which the index partitions are scanned for a previous fetch request is a same direction as the direction indicated in a current fetch request, wherein the direction indicated in the fetch request is capable of having been modified; and if the fetch direction for the previous fetch request and direction indicated in the current fetch request are different, then discarding all saved nodes for the index partitions and selecting one node from a last selected node.

Claims 6-10 provide further limitations concerning the direction of the fetch request and index partitions. The Examiner cited other sections of Cromwell as teaching the additional requirements of these claims that concern how to fetch in different directions in a result table to retrieve a row from a result table. Nowhere does the cited Cromwell teach or suggest the additional requirements of these claims providing additional requirements concerning index partitions on table partitions. Instead, the cited Cromwell discusses how to fetch forward or backward through a result table whose sequential rows are on multiple pages.

Claim 11 depends from claim 1 and further requires discarding the cached keys if the fetch request is in an opposite direction of a previous fetch request; determining a new set of nodes from each index partition; and caching the determined new set of nodes when performing the fetch operation.

The Examiner cited para. [0097] of Cromwell as teaching discarding the cached keys if the fetch request is in an opposite direction of a previous fetch request. (FOA, pg. 8) Applicants traverse.

The cited para. [0097] discusses how the data manger fetches a number of pages from storage. The database program uses a statistical consideration to determine whether rows are being sequentially accessed in order to determine whether to prefetch improve performance. Although the cited para. [0097] discusses using a statistical algorithm to determine whether to prefetch for sequential access, the Examiner has not cited any part of para. [0097] that teaches or suggests discarding cached keys if the fetch request is in an opposite direction of a previous fetch request. The cited para. [0097] discusses detecting sequential access. However, there is no mention or teaching of the claim requirement of discarding cached keys if the fetch request is in an opposite direction of the previous fetch request.

Accordingly, claim 11 provides additional grounds of patentability over the cited art because the additional requirements of these claims are not taught or suggested in the cited art.

2. Added Claims 37-59

Added claims 37-47 substantially include the requirements of pending claims 1 and 3-12 in system form and claims 48-59 substantially include the requirements of claims 1, 3-12, and 31 in "article of manufacture" form.

The preamble and additional requirements of system claim 37 are found in FIG. 1a and corresponding discussion in the Specification and in the original canceled system claim 13.

The preamble of claim 48 is disclosed in at least para. 46 of the Specification and original canceled claim 19.

Applicants submit that claims 37-59 are patentable over the cited art because they substantially include the requirements of claims 1, 3-12, and 31, which are patentable over the cite dart for the reasons discussed above.

Conclusion

For all the above reasons, Applicant submits that the pending claims 1, 3-12, 31, 32, and 37-59 are patentable over the art of record. Should any additional fees be required, please charge Deposit Account No. 09-0460.

The attorney of record invites the Examiner to contact him at (310) 553-7977 if the Examiner believes such contact would advance the prosecution of the case.

Dated: February 2, 2009

By: /David Victor/

David W. Victor
Registration No. 39,867

Please direct all correspondences to:

David Victor
Konrad Raynes & Victor, LLP
315 South Beverly Drive, Ste. 210
Beverly Hills, CA 90212
Tel: 310-553-7977
Fax: 310-556-7984